

TP10 – Contours actifs

Un *contour actif* (ou *snake*) est un modèle de courbe déformable qui évolue au cours du temps pour se fixer sur les contours, ce qui peut aider l'utilisateur à segmenter les objets visibles dans une image. Ce modèle est très utilisé, notamment, en imagerie médicale (segmentation de tumeurs, etc.).

Le contour actif est représenté par une courbe paramétrique $P(s) = [x(s), y(s)]^T \in \mathbb{R}^2$, où $s \in [0, 1]$ est une abscisse curviligne. Pour forcer la courbe à être fermée, on impose $P(1) = P(0)$. Le but est d'obtenir une courbe qui « colle » aux contours de l'image \mathcal{I} , qui soit aussi courte que possible et qui oscille le moins possible. La première de ces propriétés correspond à l'*attache aux données*, les deux autres à des termes de *régularisation*, qui sont nécessaires pour garantir une certaine robustesse au bruit et aux données manquantes. Dans le modèle original [Kass, Witkins et Terzopoulos, 1988], la régularité de la courbe est modélisée par l'*énergie interne* $E_{\text{int}}(P(s)) = \frac{1}{2} (\alpha \|P'(s)\|^2 + \beta \|P''(s)\|^2)$, où les paramètres $\alpha > 0$ et $\beta > 0$ sont choisis par l'utilisateur. Quant à l'adéquation aux contours, elle est modélisée par l'*énergie externe*, qui pénalise les faibles gradients :

$$E_{\text{ext}}(P(s)) = -\|\nabla \mathcal{I}(P(s))\|^2 \quad (1)$$

L'équation d'Euler-Lagrange associée à l'énergie $\int_0^1 [E_{\text{ext}}(P(s)) + \frac{1}{2} (\alpha \|P'(s)\|^2 + \beta \|P''(s)\|^2)] ds$ s'écrit :

$$\forall s \in [0, 1], \quad \nabla E_{\text{ext}}(P(s)) - \alpha P^{(2)}(s) + \beta P^{(4)}(s) = 0 \quad (2)$$

En notant $\nabla E_{\text{ext}} = [F_x, F_y]^T$, la solution de l'équation (2) est atteinte lorsque l'itération suivante converge :

$$\forall s \in [0, 1], \quad \begin{cases} x^{n+1}(s) = x^n(s) - \gamma [F_x(x^n(s), y^n(s)) - \alpha (x^n)^{(2)}(s) + \beta (x^n)^{(4)}(s)] \\ y^{n+1}(s) = y^n(s) - \gamma [F_y(x^n(s), y^n(s)) - \alpha (y^n)^{(2)}(s) + \beta (y^n)^{(4)}(s)] \end{cases} \quad (3)$$

où $\gamma > 0$ est un paramètre fixé par l'utilisateur. Pour implémenter cette itération, on utilise les schémas de différences finies suivants ($h \ll 1$) :

$$\begin{cases} (P^n)^{(2)}(s) = P^n(s+h) - 2P^n(s) + P^n(s-h) \\ (P^n)^{(4)}(s) = P^n(s+2h) - 4P^n(s+h) + 6P^n(s) - 4P^n(s-h) + P^n(s-2h) \end{cases} \quad (4)$$

En notant $\mathbf{x} = [x(0), x(h), \dots, x(1-h)]^T$ et $\mathbf{y} = [y(0), y(h), \dots, y(1-h)]^T$ la discrétisation de la courbe, et en remplaçant les opérateurs de dérivation par leurs approximations discrètes (4), l'itération (3) se réécrit :

$$\begin{cases} \mathbf{x}^{n+1} = \mathbf{A} \mathbf{x}^n + \mathbf{B}_x(\mathbf{x}^n, \mathbf{y}^n) \\ \mathbf{y}^{n+1} = \mathbf{A} \mathbf{y}^n + \mathbf{B}_y(\mathbf{x}^n, \mathbf{y}^n) \end{cases} \quad (5)$$

Dans (5), $\mathbf{A} = \mathbf{I} + \gamma (\alpha \mathbf{D}_2 - \beta \mathbf{D}_2^T \mathbf{D}_2)$, où \mathbf{I} désigne la matrice identité et \mathbf{D}_2 est l'approximation discrète suivante de la dérivée seconde (notez les cas particuliers de la première ligne et de la dernière ligne) :

$$\mathbf{D}_2 = \begin{bmatrix} -2 & 1 & & & 1 \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & 1 & -2 & 1 \\ 1 & & & & 1 & -2 \end{bmatrix} \quad (6)$$

donc $\mathbf{D}_2^T \mathbf{D}_2$ est l'opérateur de dérivation à l'ordre 4. Enfin, $\mathbf{B}_x(\mathbf{x}^n, \mathbf{y}^n)$ et $\mathbf{B}_y(\mathbf{x}^n, \mathbf{y}^n)$ sont définis par :

$$\mathbf{B}_x(\mathbf{x}^n, \mathbf{y}^n) = -\gamma \begin{bmatrix} F_x(x^n(0), y^n(0)) \\ \vdots \\ F_x(x^n(1-h), y^n(1-h)) \end{bmatrix} \quad \text{et} \quad \mathbf{B}_y(\mathbf{x}^n, \mathbf{y}^n) = -\gamma \begin{bmatrix} F_y(x^n(0), y^n(0)) \\ \vdots \\ F_y(x^n(1-h), y^n(1-h)) \end{bmatrix} \quad (7)$$

Pour implémenter un contour actif, il faut donc se donner une forme initiale $\mathbf{P}^0 = [\mathbf{x}^0, \mathbf{y}^0]$, puis effectuer les mises à jour (5) jusqu'à ce que l'algorithme converge.

Exercice 1 : étude de l'énergie externe

Lancez le script `exercice_0`, qui affiche le champ de force $\nabla E_{\text{ext}} = [F_x, F_y]^\top$ associé à l'énergie externe (1). Pour attirer davantage le contour actif vers les contours de l'image, on peut appliquer un filtre de convolution gaussien à l'image avant de calculer son gradient, c'est-à-dire remplacer l'énergie externe (1) par :

$$E_{\text{ext}}(P(s)) = -\|\nabla (G_{\sigma,T} * \mathcal{I})(P(s))\|^2 \quad (8)$$

où $G_{\sigma,T}$ est un noyau gaussien d'écart-type σ (par exemple $\sigma = 5$) et de taille $T \times T$ (par exemple $T = 3\sigma$).

Faites une copie du script `exercice_0`, de nom `exercice_1`, que vous modifierez de manière à afficher le champ de force externe associé à cette nouvelle énergie, avec les fonctions `fspecial` et `conv2` (l'option `'same'` force l'image résultat à avoir la même taille que l'image initiale).

Exercice 2 : implémentation du contour actif

Écrivez les fonctions `calcul_A` et `iteration`, qui doivent permettre au script `exercice_2` de :

1. Calculer **A** en utilisant (6) (utilisez la fonction `spdiags` de Matlab plutôt que des boucles `for`!).
2. Initialiser le contour actif en demandant à l'utilisateur de cliquer les points de contrôle du polygone initial.
3. Faire évoluer le contour actif selon l'itération (5).

Le jeu de paramètres fourni devrait vous permettre d'obtenir des résultats acceptables sur l'image `coins.png`, qui est une image interne à Matlab. En ajustant au mieux les paramètres, essayez ensuite de détourner la tumeur sur l'image `IRM.png`. Vous observez que le contour actif a tendance à « traverser » le contour de l'objet. Pour améliorer les résultats, une solution consiste à faire diffuser l'énergie vers les contours.

Exercice 3 : diffusion vers les contours

L'amélioration des résultats passe par une modification du champ de force externe $[F_x, F_y]^\top$, qui attire le contour actif vers les contours visibles dans l'image. Un bon modèle de force externe est celui de *diffusion vers les contours* (GVF, pour *Gradient Vector Flow*, cf. [Xu et Prince, 1998]), défini comme le minimiseur (argmin) de la fonctionnelle suivante :

$$\iint_{x,y} \{ \|\nabla E_{\text{ext}}^0(x,y)\|^2 \|[F_x, F_y]^\top(x,y) - \nabla E_{\text{ext}}^0(x,y)\|^2 + \mu_{\text{GVF}} [\|\nabla F_x(x,y)\|^2 + \|\nabla F_y(x,y)\|^2] \} dx dy \quad (9)$$

où $\nabla E_{\text{ext}}^0 = [F_x^0, F_y^0]^\top$ désigne le champ de force externe **sans filtrage gaussien**. D'après le calcul des variations, l'équation d'Euler-Lagrange associée s'écrit, en un point (x,y) quelconque (Δ désigne le laplacien) :

$$\begin{cases} \|\nabla E_{\text{ext}}^0(x,y)\|^2 [F_x(x,y) - F_x^0(x,y)] - \mu_{\text{GVF}} \Delta F_x(x,y) = 0 \\ \|\nabla E_{\text{ext}}^0(x,y)\|^2 [F_y(x,y) - F_y^0(x,y)] - \mu_{\text{GVF}} \Delta F_y(x,y) = 0 \end{cases} \quad (10)$$

L'itération suivante permet de résoudre les équations (10), appelées *équations de diffusion généralisées* :

$$\begin{cases} F_x^{n+1}(x,y) = F_x^n(x,y) - \gamma_{\text{GVF}} \{ \|\nabla E_{\text{ext}}^0(x,y)\|^2 [F_x^n(x,y) - F_x^0(x,y)] - \mu_{\text{GVF}} \Delta F_x^n(x,y) \} \\ F_y^{n+1}(x,y) = F_y^n(x,y) - \gamma_{\text{GVF}} \{ \|\nabla E_{\text{ext}}^0(x,y)\|^2 [F_y^n(x,y) - F_y^0(x,y)] - \mu_{\text{GVF}} \Delta F_y^n(x,y) \} \end{cases} \quad (11)$$

Faites une copie du script `exercice_0`, de nom `exercice_3`, que vous modifierez de manière à afficher le champ de force externe GVF, qui peut être calculé en 300 pas de l'itération (11). Utilisez la fonction `del2` pour calculer le laplacien. Donnez les valeurs suivantes aux paramètres du modèle : $\gamma_{\text{GVF}} = 0,01$ et $\mu_{\text{GVF}} = 2$.

Lancez le script `exercice_3` sur l'image `pears.png`, qui est relativement difficile à segmenter (attention, le calcul du champ de force externe est un peu long). Relancez ensuite le script `exercice_2`, en essayant de segmenter différents fruits : que pensez-vous des résultats ? Vous pouvez maintenant tester cette méthode de segmentation sur d'autres images, en veillant à ajuster ses (nombreux) paramètres au cas par cas.